18/7/2014

Universidad Nacional Experimental Del Táchira (UNET)
Decanato De Docencia
Departamento de Ingeniería Informática
Asignatura: Comunicaciones 1

PROTOCOLOS HTTP Y HTTPS

Autores:
Jessica Ramírez C.I: 21.219.949
Tyson Cardelli C.I: 23.542.402
Carlos Rangel C.I: 21.003.721

Índice

1. Hypertext Transfer Protocol
    1.1. Características
    1.2. Usos de HTTP
        1.2.1. Ejemplo de Uso
2. Hypertext Transfer Protocol Secure (HTTPS)
    2.1. Características
    2.2. Usos del HTTPS
    2.3. Configuración del servidor
        2.3.1. Usar un control de acceso
    2.4. Ejemplos de usos de HTTPS
        2.4.1. En navegadores
        2.4.2. En dispositivos móviles
3. Diferencias con HTTP
4. Referencias
5. Glosario

Hypertext Transfer Protocol

HTTP son las siglas en inglés de ***HiperText Transfer Protocol*** (en español, **protocolo de transferencia de hipertexto**). Es un protocolo de red (en informática un protocolo se puede definir como un conjunto de reglas a seguir) para publicar páginas de web o HTML. HTTP es la base sobre la cual está fundamentado Internet, o la WWW.

HTTP fue desarrollado por el World Wide Web Consortium y la Internet EngineeringTaskForce, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores,proxies) para comunicarse

Características

El protocolo HTTP funciona a través de solicitudes y respuestas entre un cliente (por ejemplo un navegador de Internet) y un servidor (por ejemplo la computadora donde residen páginas web). A una secuencia de estas solicitudes se le conoce como sesión de HTTP.

La información que el navegador de Internet está presentando en un momento dado, se identifica en la llamada "barra de navegación", que comienza con http y se le conoce como URI (más conocido como URL).

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "useragent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente

mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

Usos de HTTP

HTTP es un protocolo usado para cada transacción de World Wide Web, por lo cual es el protocolo más importante usado por medio de la Internet

Para que un cliente pueda realizar una conexión con un servidor se debe de especificar el DNS o dirección IP de dicho servidor. Está dirección HTTP se denomina URL y se compone de las siguientes partes:

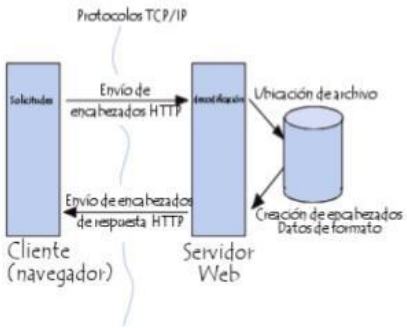
[http://]hostname [:] puerto [/] recurso

donde *hostname* es el nombre de dominio o una dirección IP, *puerto* en el número del puerto al que se le envía la petición (si no especifica por defecto es el **80**) y *recurso* especifica el **nombre del recurso** que se indica (si no indica un recurso se asume que es "/").

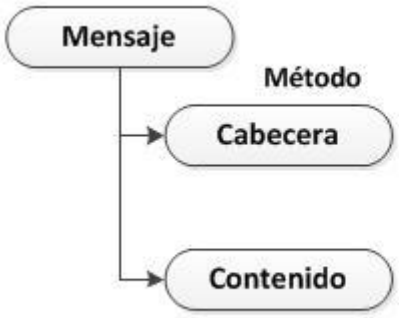
La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

El navegador realiza una solicitud HTTP

El servidor procesa la solicitud y después envía una respuesta HTTP



Durante una transacción HTTP se envían mensajes, a estos mensajes se le conocen como **peticiones requests,** estas peticiones tienen una estructura y contiene los siguientes elementos: una cabecera y opcionalmente un contenido. La cabeceras incluyen como primera línea el método que queremos minvocar (GET, POST, etc.)



Cabeceras Generales

Cache-Control, son directivas que se han de tener en cuenta a la hora de mantener el contenido en una caché.
Connection, permite especificar opciones requeridas para una conexión.
Date, representa la fecha y la hora a la que se creó el mensaje.
Pragma, usado para incluir directivas de implementación.
Transfer-Encoding, indica la codificación aplicada al contenido.
Upgrade, permite al cliente especificar protocolos que soporta.
Via, usado por pasarelas y proxies para indicar los pasos seguidos.

Las cabeceras generales son las que se aplican tanto a peticiones como a respuestas, pero no al contenido que se transmite. Las cabeceras de petición permiten al cliente pasar información al servidor sobre la petición y sobre el cliente. Las cabeceras de entidad permiten definir información adicional sobre el contenido que se transmite y en caso de que no haya contenido, sobre el recurso al que se quiere acceder con la petición.

Un uso frecuente de HTTP en Windows es para transmitir por secuencias contenido desde un codificador a un servidor de Windows Media, con objeto de distribuir secuencias entre equipos que ejecuten versiones diferentes de Servicios de Windows Media o equipos que estén separados por un firewall, así como para descargar listas de reproducción generadas dinámicamente desde un servidor web. HTTP es especialmente útil para aquellos clientes que reciben contenidos transferidos por secuencias a través de un firewall, puesto que este protocolo normalmente está configurado para usar el puerto 80, que la mayoría de los firewall no bloquean.

Ejemplo de Uso

Para obtener un recurso con el URL http://www.example.com/index.html

1. Se abre una conexión al host www.example.com, puerto 80 que es el puerto por defecto para HTTP.

2. Se envía un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
  .
  .
  .
</body>
</html>
```

HTTP define 8 métodos (algunas veces referido como "verbos") que indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa, si los datos pre-existentes o datos que se generan de forma dinámica, depende de la aplicación del servidor. A menudo, el recurso corresponde a un archivo o la salida de un ejecutable que residen en el servidor.

HEAD

Pide una respuesta idéntica a la que correspondería a una petición GET, pero sin el cuerpo de la respuesta. Esto es útil para la recuperación de meta-información escrita en los encabezados de respuesta, sin tener que transportar todo el contenido.

GET

Pide una representación del recurso especificado. Por seguridad **no deberías**er usado por aplicaciones que causen efectos ya que transmite información a través de la URI agregando parámetros a la URL.

Ejemplo:

GET /images/logo.png HTTP/1.1 obtiene un recurso llamado logo.png

Ejemplo con parámetros:

/index.php?page=main&lang=es

POST

Envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas.

PUT

Sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor, esto es porque en POST utiliza un **mensaje multiparte** y el mensaje es decodificado por el servidor. En contraste, el método PUT te permite escribir un archivo en una conexión socket establecida con el servidor.

La desventaja del método PUT es que los servidores de hosting compartido no lo tienen habilitado.

Ejemplo:

PUT /path/filename.html HTTP/1.1

DELETE

Borra el recurso especificado.

TRACE

Este método solicita al servidor que envíe de vuelta en un mensaje de respuesta, en la sección del cuerpo de entidad, toda la data que reciba del mensaje de solicitud. Se utiliza con fines de comprobación y diagnóstico.

OPTIONS

Devuelve los métodos HTTP que el servidor soporta para un URL específico.Esto puede ser utilizado para comprobar la funcionalidad de un servidor web mediante petición en lugar de un recurso específico.

CONNECT

Se utiliza para saber si se tiene acceso a un host, no necesariamente la petición llega al servidor, este método se utiliza principalmente para saber si un proxy nos da acceso a un host bajo condiciones especiales, como por ejemplo "corrientes" de datos bidireccionales encriptadas (como lo requiere SSL).

Hypertext Transfer Protocol Secure (HTTPS)

Hypertext Transfer ProtocolSecure (en español: Protocolo seguro de transferencia de hipertexto), conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

El HTTPS fue creado por Netscape Communications en 1994 para su navegador Netscape Navigator.

Es una combinación del protocolo HTTP y protocolos criptográficos. Se emplea para lograr conexiones más seguras en la WWW, generalmente para transacciones de pagos o cada vez que se intercambie información sensible (por ejemplo, claves) en internet.

De esta manera la información sensible, en el caso de ser interceptada por un ajeno, estará cifrada.

El nivel de protección que ofrece depende de la corrección de la implementación del navegador web, del software y de los algoritmos criptográficos soportados. Además HTTPS es vulnerable cuando es aplicado a contenido estático públicamente disponible.

Características del HTTPS

Para distinguir una comunicación o página web segura, la URL debe comenzar con "https://" (empleando el puerto 443 por defecto); en tanto la tradicional es "http://" (empleando el puerto 80 por defecto).

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

HTTPS fue adoptado como estándar web por el grupo IETF tras la publicación del RFC 2818 en mayo de 2000.

HTTP opera en la capa más alta del modelo OSI, la capa de aplicación; pero el protocolo de seguridad opera en una subcapa más baja, cifrando un mensaje HTTP previo a la transmisión y descifrando un mensaje una vez recibido. Estrictamente hablando, HTTPS no es un protocolo separado, pero refiere el uso del HTTP ordinario sobre una Capa de Conexión Segura cifrada Secure Sockets Layer (SSL) o una conexión con Seguridad de la Capa de Transporte (TLS).

Usos del HTTPS

Https usa cifrado para proteger el tráfico de Internet con el fin de impedir que otros usuarios de la red puedan espiarlo o alterarlo.

Cuando utilizamos http la información que mandamos y recibimos a través del navegador realiza una serie de saltos entre diferentes routers que se encuentran entre el cliente (nuestro navegador web) y el servidor web remoto. En cada uno de estos saltos la información puede retransmitirse a todos los dispositivos (PCs, servidores, etc.) conectados a la red del router en cuestión. No podemos evitar esta retransmisión a todos los dispositivos de la red, con el riesgo de que esta información pueda ser interceptada desde

cualquiera de estos dispositivos, pero si podemos encriptar esta información utilizando https que añada a la información enviada cifrado SSL para que aunque se intercepte esta información (contraseñas, datos bancarios, etc.), no pueda ser descifrada y por tanto utilizada. Https usa los protocolos de Capa de sockets seguros (SSL) o Seguridad de la capa de transporte (TLS) para proteger la información. El uso de https está especialmente recomendado cuando se usan redes wifi no seguras, pero su uso en todo tipo de redes es siempre positivo.

Configuración del servidor

Para preparar un servidor web que acepte conexiones HTTPS, el administrador debe crear un certificado de clave pública para el servidor web. Este certificado debe estar firmado por una autoridad de certificación para que el navegador web lo acepte. La autoridad certifica que el titular del certificado es quien dice ser. Los navegadores web generalmente son distribuidos con los certificados raíz firmados por la mayoría de las autoridades de certificación por lo que estos pueden verificar certificados firmados por ellos.

Usar un control de acceso

El sistema puede también ser usado para la de clientes con el objetivo de limitar el acceso a un servidor web a usuarios autorizados. Para hacer esto el administrador del sitio típicamente crea un certificado para cada usuario, un certificado que es guardado dentro de su navegador. Normalmente, este contiene el nombre y la dirección de correo del usuario autorizado y es revisado automáticamente en cada reconexión para verificar la identidad del usuario, potencialmente sin que cada vez tenga que ingresar una contraseña.

Ejemplos de usos de HTTPS

En navegadores:

Cada vez más servicios de internet permiten el uso de https, aparte de la mayoría de entidades que prestan servicios financieros y sitios de comercio electrónico que lo incorporan de forma permanente, existen otros servicios como Gmail que también lo incorporan de forma automática y permanente. Así mismo existen servicios como Google,

Facebook, Twitter, Hotmail que permiten navegar de forma segura utilizando https pero no lo incorporan de forma predeterminada.

También existen extensiones para algunos de los navegadores más populares que fuerzan la navegación segura (https mediante cifrado SSL o TLS), a continuación se citan algunas de estas extensiones:

- ✓ Firefox: utilizar extensiones Force-Tls o HTTPS-Everywhere.
- ✓ Google Chrome: utilizar extensión KB SSL Enforced, una vez descargado de Internet e instalado, fuerza conexiones seguras en todos los sitios web que es posible.
- ✓ Opera: utilizar extensión RedirecttoHttps, una vez descargado de Internet e instalado, fuerza conexiones seguras en todos los sitios web que es posible.

HTTPS Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

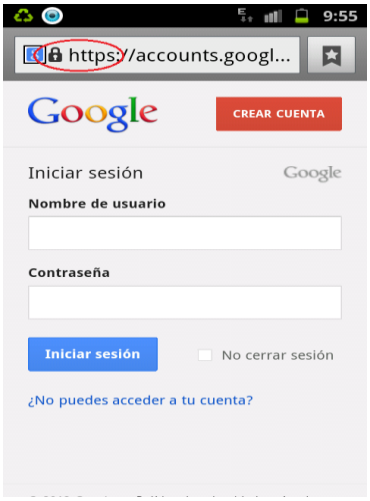
En dispositivos móviles:

Dado que en los dispositivos móviles se ejecutan prácticamente las mismas acciones que en una computadora de escritorio, es necesario conocer las características de los navegadores en este tipo de dispositivos para estar seguros de que se navega por sitios seguros. Al utilizar ya sea smartphones o tablets para hacer compras en línea, revisar el correo electrónico, navegar en las redes sociales, consultar el estado de la cuenta bancaria y muchas otras opciones; el usuario está expuesto a amenazas como el robo de información, el phishing o la infección con códigos maliciosos.

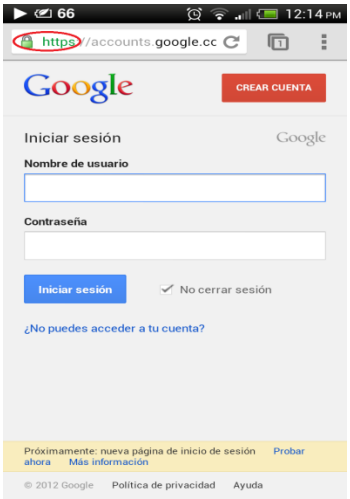
Después de tener instalada en el dispositivo una solución de seguridad, que brinde protección contra códigos maliciosos y que además ofrezca características adicionales como funciones de firewall o antispam incluso sobre mensajes SMS y MMS como ESET Mobile Security.

De esta forma, al estar navegando utilizando dispositivos móviles es importante conocer si al ingresar a un sitio web este utiliza el protocolo seguro de navegación HTTPS. Por ejemplo los dispositivos con sistema operativo Android traen por defecto un navegador,

que una vez se ingresa a un sitio que utiliza el protocolo de navegación segura (HTTPS) muestra en la parte izquierda un candado e indica el tipo de protocolo utilizado.



Por ejemplo: Google Chrome en su versión móvil, muestra al lado izquierdo de la barra de direcciones y en un color diferente tanto un candando como la sigla **https** para los sitios que utilizan este protocolo para el intercambio de información. En la siguiente captura se puede observar lo descrito:



Usar protocolo HTTPS brinda más privacidad y seguridad que una conexión web sin encriptación. Disminuye el riesgo de que terceros intercepten y usen indebidamente la información. Muchos visitantes de sitios se sienten más cómodos al realizar pagos y compartir información personal cuando sabe que están usando una conexión con SSL

Diferencias con HTTP

En el protocolo HTTP las URLs comienzan con "http://" y utilizan por defecto el puerto 80, las URLs de HTTPS comienzan con "https://" y utilizan el puerto 443 por defecto.

HTTP es inseguro y está sujeto a ataques man-in-the-middle y eavesdropping que pueden permitir al atacante obtener acceso a cuentas de un sitio web e información confidencial. HTTPS está diseñado para resistir esos ataques y ser más seguro.

Referencias

http://informatica.uv.es/iiguia/IST/Tema2.pdf

http://es.slideshare.net/babp/protocolo-https-3617275

http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html

http://www.desarrolloweb.com/articulos/protocolo-http-ftp.html

Glosario

Capa de sockets seguros (SSL): El mecanismo de Capa de sockets seguros (SSL) estándar del sector, que utiliza certificados digitales para la autenticación, se puede utilizar para la comunicación segura en despliegue de TivoliProvisioning Manager.

TransportLayer Security (TLS): En español «seguridad de la capa de transporte») son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

URL: Son las siglas en inglés de uniform resource locator (en español localizador uniforme de recursos), que sirve para nombrar recursos en Internet. Este nombre tiene un formato estándar y tiene como propósito asignar una dirección única a cada uno de los recursos disponibles en Internet, com por ejemplo textos, imágenes, vídeos, etc.

Cookie (o galleta informática): Es una pequeña información enviada por un sitio web y almacenado en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

World Wide Web (WWW) o Red informática mundial: Conocida como la web, es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

Domain Name System o DNS (en español: sistema de nombres de dominio): Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes.

Ataque Man-in-the-middle: Ataque man-in-the-middle o JANUS (MitM o intermediario, en español) es un ataque en el que el enemigo adquiere la capacidad de leer, insertar y modificar a voluntad, los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado.

Eavesdropping: Es un proceso mediante el cual un agente capta información - en claro o cifrada - que no le iba dirigida. Aunque es en principio un ataque completamente pasivo, lo más peligroso del eavesdropping es que es muy difícil de detectar mientras que se produce.